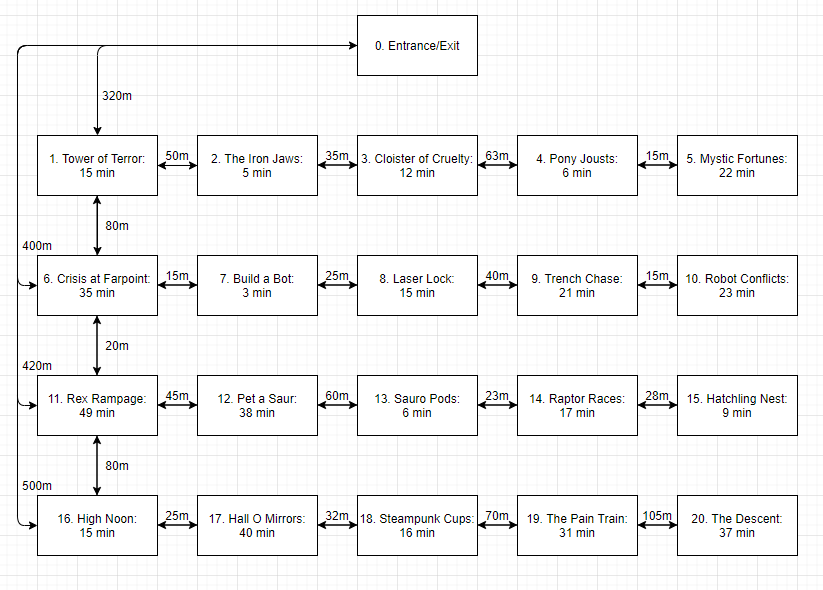
# Implementation Report – Step3

For step 3, I used an edge list graph to generate my graph with all the ride options as vertices and then added the edges with their relevant weight (which represented distance in this case).

The diagram below represents the map of the park, the distances between the rides, and the wait time at each ride.



I used a djikstra path to then calculate the distance to each ride from the entrance and displayed this appropriately.

For the second part of step 3, I was asked to implement and display the distance from entrance to rides recommended from step 2. So I stored the “collatedRides” collected from step 2, which identifies appropriate rides for the selected party, and then passed this in to the path generator which displays only the information of the parties rides.

I struggled to get this part working and spent most of my time fixing this part up. Initially I had built a shortest path algorithm but this was retracted from the product specification so I decided to utilise the material I had already implemented to fulfil this section of the assignment